

## **АРХИТЕКТУРА И ИНВАРИАНТНОЕ ЯДРО СИСТЕМ ПОДДЕРЖКИ ВЫРАБОТКИ РЕШЕНИЙ НАЧАЛЬНОГО УРОВНЯ**

**Р.Ф. Хабибуллин, С.М. Шишкин**

В работе рассматривается простой вариант архитектуры систем поддержки выработки решений (СПВР, см. [1, 2]), обладающий, тем не менее, достаточно развитыми возможностями и средствами, и инвариантное ядро программного обеспечения СПВР.

В архитектуре СПВР начального уровня выделяются следующие группы основных пользовательских функций и команд: функции формирования и изменения вариантов решений; функции анализа, оценки и сравнения вариантов; функции работы с вариантами решений и дополнительные и служебные функции и команды.

В системах поддержки выработки решений начального уровня имеется одна рабочая область конечного пользователя (КП) - лица, принимающего решение (ЛПР), или аналитика, в которой содержатся все варианты решений, доступные пользователю. Рабочая область включает в себя один текущий, один опорный и произвольное число запомненных вариантов решения.

Текущий вариант решения характеризуется тем, что только к нему конечный пользователь может применять функции формирования и изменения вариантов решения, в результате которых текущий вариант изменяется, приобретая новое значение.

Опорный вариант решения играет роль “направляющего” решения. Это такой частичный или полный вариант решения, который является основой для анализа и рассмотрения вариантов, являющихся его изменением и дополнением. Когда конечный пользователь только начинает работу, опорный и текущий варианты совпадают. После применения функций формирования и изменения варианта решения, опорный вариант решения останется прежним, а текущий вариант будет изменяться в соответствии с совершаемыми действиями. Если конечный пользователь останется не удовлетворен результатами манипуляций с решением, он может вернуться к опорному варианту и начать все снова.

Запомненные варианты решения могут служить конечному пользователю для сохранения в них других перспективных для дальнейшей работы вариантов решения, вариантов претендующих на роль окончательного решения и т.д. Каждый запомненный вариант получает имя, которое задает

КП при его сохранении, и которое используется при выборе варианта при его активизации, сравнении и т.д.

Для отображения вариантов решения в процессе их формирования, отображения их особенностей и характеристик в СПВР используются одна или несколько форм представления решения. Формы представления решения (ФПР) служат для визуального отображения выбранного полного или частичного варианта решения с возможностью доступа к значениям всех компонент решения, выбора и присвоения им значений или изменения уже присвоенных значений. Все произведенные изменения в рассматриваемом варианте решения немедленно отображаются в ФПР, таким образом, ЛПР в той или иной форме может наблюдать за динамикой изменений рассматриваемого решения в процессе манипулирования с ним. В зависимости от рассматриваемой проблемы принятия решения и конкретных условий, в которых она решается, могут быть разработаны и использоваться несколько различных ФПР, благодаря этому пользователь может с разных сторон воспринимать вариант решения. Каждая ФПР может отображаться в одной или нескольких экранных формах, в последнем случае механизм листания и выбора экранных форм ФПР закладывается в саму ФПР. Нужно отметить, что все формы представления решения фиксируются при разработке СПВР. В процессе работы КП может переходить от использования одной формы к использованию другой.

С каждым вариантом решения связано *Примечание*, в котором могут содержаться значения тех или других атрибутов этого варианта решения (например, значения некоторых характеристик варианта решения, дата и время его создания, дата и время последней работы с ним и т.д.), а также пометки пользователя. В простейшем случае примечание является текстовым полем произвольного размера, в которое КП может заносить свои пометки. В общем случае формат записи примечания определяется разработчиком СПВР. Примечание служит для краткой характеристики варианта решения, которая может оказаться полезной КП при выборе варианта для работы, а также для того, чтобы пользователю легче было вспомнить, чем он руководствовался при разработке этого варианта, его текущую субъективную оценку этого варианта.

В основном режиме работы конечному пользователю (КП) непосредственно доступны все операции манипулирования с решением и обращение к остальным основным функциям и командам СПВР. После вызова соответствующих функций или команд в режиме их выполнения доступны те операции и команды, которые определяются смыслом выполняемой функции. После выполнения соответствующих действий либо процесс будет возвращен в основной режим автоматически, либо по команде пользователя. Пользователь может, рассматривая текущий вариант решения в любой из предусмотренных форм представления решения, изменять его как с по-

мощью команд непосредственного выбора и изменения значений компонент, так и с использованием интеллектуальных средств - достраивателей и улучшателей. При этом, помечивая отдельные компоненты, он может защитить их от неконтролируемого изменения. Пользователь может оценить текущий вариант решения и его характеристики в любой из задействованных форм оценивания и сравнить его с одним или несколькими другими вариантами решения в любой из задействованных форм сравнения. Для работы с совокупностью выработанных вариантов решения пользователю предоставляются команды сохранения текущего варианта решения, активизации запомненного варианта решения, возврата к опорному варианту решения, придания текущему варианту статуса опорного и др. Пользователь может также при необходимости использовать функцию просмотра вариантов решений, которая служит для общего анализа того, что разработано.

### **Функции формирования и анализа текущего варианта решения**

С помощью команды *Выбрать ФПР* выбирается форма представления текущего варианта решения. При вызове этой команды СПВР представит перечень обозначений (например, имен) всех ФПР, которые определены в системе. КП может выбрать любую из них. Выбранная ФПР становится текущей и выполнение команды завершится возвратом в основной режим и представлением текущего варианта решения в выбранной ФПР.

Команды непосредственного манипулирования значениями компонент текущего варианта решения, например, *Выбрать компоненту*, *Установить значение*, *Снять значение* и др., полностью определяются рассматриваемой в СПВР проблемой, поэтому обращение к ним в каждой из форм представления решения может быть определено по-разному в каждой конкретной реализации СПВР. После выполнения каждой команды манипулирования текущий вариант решения изменяется и отображается в текущей ФПР.

В СПВР определены также средства, которые позволяют пользователю помечать значения компонент текущего варианта решения с целью временного запрещения изменения их значений. Функция помечивания поддерживается, вообще говоря, несколькими различными командами (например, *выбрать значение*, *пометить значение*, *пометить все установленные значения*, *снять пометку*, *снять все пометки*). Обращение к этим командам так же, как и в случае использования функций непосредственного манипулирования, полностью определяется формой представления текущего варианта решения. Если КП попытается изменить ранее помеченное значение компоненты текущего решения, то СПВР предупредит, что данное значение компоненты помечено и изменить его невозможно до тех пор, пока

не будет снято это помечивание.

Интеллектуальные средства изменения текущего варианта решения автоматически по формализованным методам формируют варианты решения, удовлетворяющие, по возможности, заданным свойствам. В данной архитектуре определены два типа интеллектуальных средств. Это достраиватели и улучшатели.

Достраиватели, не изменяя уже установленных значений компонент текущего частичного варианта решения, вырабатывают значения для остальных его компонент так, чтобы полученный вариант обладал, по возможности, наилучшими в определенном смысле значениями характеристик. Для использования достраивателей в СПВР определена команда *Достроить*. При вызове этой команды КП может выбрать любой из представленных достраивателей и запустить процесс его выполнения. В процессе достраивания СПВР может некоторым образом показывать состояние процесса. После завершения работы достраивателя выработанный им вариант решения становится текущим, представляется в текущей ФПР, и КП возвращается к работе над новым текущим решением.

Улучшатели, отталкиваясь от текущего варианта решения и корректируя его, отыскивают новый вариант так, чтобы он обладал, по возможности, улучшенными характеристиками. При этом, если какие-то значения компонент текущего решения помечены как временно запрещенные к изменению, то они не могут быть изменены улучшателем. Благодаря помеченным компонентам, КП может гибко разделять текущий вариант решения на ту часть, которую по тем или другим причинам не следует изменять, и часть, в которой допускается варьирование с целью улучшения решения. Использование команды *Улучшить* аналогично использованию команды *Достроить*. При этом КП выбирает нужный улучшатель, который затем выполняется.

Рассмотрим теперь функции, связанные с анализом текущего варианта решения.

Функция оценивания позволяет КП оценить в некоторой форме текущий вариант решения. Формы оценивания (ФО) служат для визуализации особенностей решения, представления значений его расчетных характеристик как скалярных, так и их динамики или пространственного распределения, оценок их значений (предельных, средних и т.д.), а также представления результатов модельных испытаний, имитационного моделирования и т.д. - всего того, что представляет или может представлять ценность для КП и может быть получено на компьютере для всесторонней и по возможности полной оценки решения с точки зрения его эффективности и приемлемости для окончательного решения конкретной проблемы. В зависимости от рассматриваемой проблемы и конкретных условий, в которых она решается, могут быть разработаны и использоваться несколько различных

ФО, каждая из которых может отображать свои конкретные особенности оцениваемого варианта решения. ФО отображается в одной или нескольких экранных формах, и в последнем случае СПВР предоставляет КП механизм “листания” и выбора экранных форм ФО.

При вызове команды *Оценить* КП может выбрать любую из ФО и запустить процесс, выполняющий формирование представления результатов оценивания текущего варианта решения по выбранной ФО. После того, как все необходимые характеристики будут вычислены, результаты оценивания будут представлены КП так, как определено выбранной ФО. Однако система продолжает оставаться в состоянии выполнения функции оценивания (команда *Оценить* не завершена) для работы с этой ФО (листание экранных форм и т.д.). Дальше КП может оценить текущий вариант решения по другой форме или выйти из функции оценивания в основной режим к текущему варианту решения и его представлению в ФПР, которые были до обращения к функции оценивания.

Функция сравнения позволяет КП сравнить текущее решение с любыми ранее выработанными и запомненными вариантами решений в некоторой форме. Формы сравнения (ФС) служат для визуализации и отображения информации, помогающей сравнить, выявить сходство и различие в самих сравниваемых вариантах и/или значениях их характеристик. Формы сравнения, служащие для представления детального (например, покомпонентного) сравнения текущего варианта решения с другим выбранным вариантом решения из рабочей области дают возможность пользователю более детально рассмотреть различия между двумя вариантами, увидеть какие именно компоненты сравниваемых решений в какой степени и в какую сторону влияют на значения характеристик. К другому типу относятся формы сравнения, предназначенные для общего сравнения текущего варианта решения с несколькими другими вариантами решения (групповое сравнение). В этих формах могут представляться как сами сравниваемые варианты решений, так и значения их характеристик и атрибутов. В СПВР может быть определено несколько различных форм индивидуального и группового сравнения и каждая из них может отображаться в одной или нескольких экранных формах.

При обращении по команде *Сравнить* к функции сравнения КП выбирает нужную ФС и те варианты, с которыми он желал бы сравнить текущий вариант решения (*Выбор сравнения*), и запускает процесс, выполняющий формирование представления результатов сравнения текущего варианта решения с выбранными вариантами решений по соответствующей ФС. После того, как будут вычислены все необходимые характеристики, результаты сравнения будут представлены КП так, как определено выбранной ФС. Однако система продолжает оставаться в состоянии (режиме) выполнения функции сравнения. Дальше КП может воспользоваться ко-

командой *Выбор сравнения* и сравнить текущий вариант решения с другими вариантами и/или по другой ФС, или командой *Выход* для выхода из функции сравнения в основной режим к текущему варианту решения и его представлению в ФПР.

### **Функции работы с вариантами**

Команда *Сохранить* предназначена для сохранения текущего варианта решения в качестве запомненного варианта и доступна только в основном режиме. СПВР предложит ввести имя сохраняемого варианта решения и сохранит точную копию текущего варианта решения.

Команда *Активизировать* предназначена для возврата к работе над ранее сохраненным запомненным вариантом. Эта функция доступна только в основном режиме. При ее вызове КП выбирает некоторый запомненный вариант решения, и тогда текущему варианту решения присваивается значение выбранного запомненного варианта. После этого функция активизации заканчивает свою работу и СПВР представит новый текущий вариант решения в текущей ФПР.

Команда *Обменять* предназначена для возврата к работе над ранее сохраненным запомненным вариантом с сохранением текущего варианта решения в этом запомненном варианте. Доступна только в основном режиме. При ее вызове КП выбирает некоторый запомненный вариант решения, и текущий вариант решения приобретает значение выбранного запомненного варианта, который, в свою очередь, приобретает значение текущего варианта решения. После этого функция обмена заканчивает свою работу и СПВР представит новое текущее решение в текущей ФПР.

Команда *Сделать опорным* текущий вариант решения может быть использована КП, например, в случае, если после некоторых изменений текущего варианта решения полученный вариант окажется с его точки зрения именно тем, который может быть принят в качестве опорного и направляющего. Команда доступна только в основном режиме. В результате ее выполнения опорный вариант решения приобретает значение текущего варианта решения.

Команда *Вернуться к опорному* предназначена для возврата к работе над опорным решением и может быть использована КП, например, в случае, если после некоторого изменения текущего варианта решения, полученный вариант не будет удовлетворять КП. Эта команда доступна только в основном режиме. В результате выполнения команды текущий вариант решения приобретает значение опорного и представляется в текущей ФПР.

Команда *Вернуться к прежнему (предыдущему) текущему* варианту решения предназначена для возврата к работе над тем текущим вариантом решения, который был до последнего его изменения какой-либо из команд

*Достроить, Улучшить, Активизировать, Обменять, Вернуться к опорному.* Команда доступна только в основном режиме. В результате выполнения команды текущий вариант решения приобретает свое предыдущее значение и представляется в текущей ФПР.

Основное применение команды *Вернуться к прежнему текущему* - после использования достраивателей и улучшателей. КП может сначала рассмотреть и проанализировать вариант решения, полученный, например, одним достраивателем, в различных ФПР, оценить его, сравнить с другими вариантами решений, при желании сохранить его, а затем вернуться с помощью этой команды к исходному текущему варианту решения, чтобы попробовать использовать другой достраиватель. В частности, таким образом КП может попробовать поочередно применить все достраиватели или улучшатели к текущему варианту решения, сохраняя при желании выработанные ими варианты решений, а затем сравнить их между собой с целью выбора лучшего варианта решения для дальнейшей работы с ним.

Команда *Отказ* в общем случае отменяет вызов, действие или результат последней заданной команды, функции или операции, неважно выполнена она или находится в процессе выполнения, и возвращает к состоянию перед ней. Эта команда доступна всегда, в любой момент работы (кнопочный отказ), и переводит состояние СПВР в состояние, бывшее до применения последней пользовательской функции или команды (включая значения вариантов решений). Точное действие команды *Отказ* оговаривается для каждой функции и команды.

Для того, чтобы КП, отвлекаясь от работы с текущим вариантом решения, мог перейти к более тщательному анализу всей совокупности ранее выработанных вариантов решения, а также для помощи при выборе запомненного варианта для активизации или обмена, предназначена функция просмотра. Обращение к функции просмотра вариантов осуществляется из основного режима или из команды *Выбор варианта* (для сохранения, активизации, обмена) с помощью команды *Просмотр вариантов*. После того, как КП выберет для просмотра некоторый вариант (запомненный, опорный или текущий), он становится активным, СПВР представит его в текущей ФПР, и КП может использовать любую из функций, доступных в режиме просмотра: *Выбор варианта* (для просмотра), *Выбор ФПР* (для активного варианта решения), *Оценить, Сравнить* (просматриваемый вариант с любым вариантом рабочей области, включая текущий), *Удалить* (заданный запомненный вариант) и команду *Выход* из режима просмотра.

Команда *Удалить* предназначена для удаления запомненного варианта решения, который активизирован в текущий момент. Команды *Выбор ФПР, Оценить* и *Сравнить* выполняются так же, как в основном режиме, с той лишь разницей, что применяются они не к текущему, а к активному варианту решения. Команда *Выход* (из просмотра) служит для возврата в

точку вызова функции просмотра, при этом восстанавливается состояние, которое было до обращения к функции просмотра (кроме удаленных вариантов).

Для работы с примечаниями в основном режиме, в режиме просмотра и из команды *Выбор варианта* доступна команда *Посмотреть примечание*. При ее вызове отображается содержание примечания текущего (в основном режиме) или активного (в режиме просмотра) вариантов решений, или выделенного варианта решения при выполнении команды *Выбор варианта*. Одновременно становится доступным редактор примечаний, с помощью которого КП может вносить в примечание дополнения и изменения.

Команда *Выход из СПВР* предназначена для завершения работы в СПВР и доступна только в основном режиме. При ее вызове СПВР представляет запрос на сохранение состояния СПВР. По команде *Сохранить состояние СПВР* все варианты решений рабочей области и состояние процесса выработки решения (текущее решение, текущая ФПР и т.д.) сохраняются для возможности восстановления при следующем обращении к системе. С помощью команды *Восстановить состояние* при входе в СПВР можно полностью восстановить ситуацию, которая была перед выходом с сохранением состояния СПВР, так что КП может продолжить работу, как будто она не прерывалась.

Кроме рассмотренных функций и команд в распоряжении конечного пользователя должны быть *Подсистема ввода* базовых исходных данных, которая обеспечивает возможность ввода всех данных, определяющих конкретную решаемую проблему и условия, в которых она решается, и *Подсистема вывода* и представления результатов, которая обеспечивает выдачу вариантов решений и значений всех необходимых характеристик, включая результаты оценки и сравнения вариантов решений, а также представление результатов работы в подходящей форме.

### **Инвариантное ядро СПВР**

Концепция СПВР предусматривает их разработку и использование для решения достаточно сложных типичных и регулярно повторяющихся практических проблем принятия решений в обыденной практике руководителей и специалистов. Разработка и создание конкретных СПВР различного функционального назначения является достаточно трудоемкой и сложной работой. Для того, чтобы уменьшить трудоемкость разработки программного обеспечения (ПО) конкретных СПВР может оказаться полезным выделить в нем общую неизменяемую для класса СПВР часть - назовем ее инвариантным ядром (ИЯ) СПВР. Основа для такого выделения есть - это независимые от конкретных проблем или общие для проблем

данного класса функции, которые должны выполняться системой. Одно ИЯ недостаточно для создания полноценной системы. Другая часть программного обеспечения, дополняющая ИЯ до полнофункциональной СПВР, будет являться преимущественно проблемно-зависимой, определяемой типом конкретной проблемы, и обеспечивать возможность конкретного функционального использования. Назовем ее функциональным наполнением (ФН) СПВР. Функциональное наполнение - это та часть программного обеспечения СПВР, которая в совокупности с ИЯ обеспечивает выполнение всех функций СПВР для поддержки решения проблемы принятия решения.

Создание инвариантного ядра позволит разработчикам СПВР сосредоточиться прежде всего на содержательной стороне проблемы принятия решения, для которой предназначена СПВР, и ее особенностях, на разработке эффективных средств, помогающих в ее решении, в частности, на разработке более совершенных дистраивателей и улучшателей, форм представления решения, средств оценки и сравнения вариантов решения и т.д.

Важная основа обеспечения возможности выделения инвариантного ядра - конкретная архитектура систем поддержки выработки решений, которая определяет совокупность конечных пользовательских функций и тем самым большую часть интерфейса конечного пользователя (ЛПР или аналитика) с системой. Эта часть интерфейса оказывается независимой от конкретной решаемой проблемы, что позволяет реализовать именно в рамках инвариантного ядра единый стандартный интерфейс взаимодействия КП с СПВР, например, в форме интегрированной среды - общей для всех решаемых проблем данного класса, и, соответственно, для всех СПВР, основанных на данной архитектуре. Благодаря этому, конечному пользователю не придется каждый раз заново привыкать к работе с новой СПВР.

Поскольку инвариантное ядро уже должно обеспечивать возможность работы с одним ФН для решения соответствующей проблемы, разумно потребовать, чтобы одно и то же ИЯ обеспечивало возможность работы (по выбору) с различными ФН для решения различных проблем.

Наконец, построение ИЯ должно в достаточной степени поддерживать возможность гибкой модификации конкретного ФН и его расширения за счет включения более удобных и эффективных средств (например, средств представления решения, оценки, формирования вариантов решения и т.д.). Благодаря этому, процесс создания, развития и совершенствования конкретной СПВР может осуществляться поэтапно, путем постепенного наращивания ее возможностей.

Таким образом, инвариантное ядро СПВР выполняет следующие основные функции:

- реализация взаимодействия с конечным пользователем СПВР, обеспечение доступа к проблемно-зависимым функциям, реализованным в ФН,

реализация остальных функций архитектуры СПВР;

- реализация взаимодействия с функциональными наполнениями для решения конкретных типов проблем;

- обеспечение гибких возможностей проектировщику и администратору системы для относительно несложной модификации и расширения состава ФН, увеличивающих функциональные возможности СПВР.

Инвариантное ядро СПВР может быть выделено различными способами, и по-разному может быть организовано его взаимодействие с ФН.

В общем случае ИЯ и ФН могут представлять собой отдельные закрытые друг от друга системы, которые взаимодействуют друг с другом через определенный информационный канал по определенному специальному протоколу. В этом случае в ФН выделяется модуль связи, который осуществляет прием данных из ИЯ, их интерпретацию в соответствии с протоколом и вызов основного модуля ФН, а также передачу данных в ИЯ. Основной модуль ФН осуществляет выполнение тех функций, которые определяются вызовом ИЯ, и подготавливает данные для передачи в ИЯ. Поддержка взаимодействия между ИЯ и ФН может осуществляться по-разному, например, через механизмы межпрограммного взаимодействия (IPC), удаленный вызов процедур (RPC), могут также использоваться при необходимости распределенные технологии типа CORBA и DCOM, или вызов удаленных Java объектов (RMI). В последних случаях фактически ИЯ взаимодействует с одной процедурой или объектом ФН, который вызывает или реализует модуль связи ФН. Отметим также, что в случае включения в СПВР нескольких ФН, это позволяет ИЯ гибко управлять каждым функциональным наполнением в отдельности.

По-видимому, этот способ позволяет разрешить многие проблемы, возникающие при разработке ИЯ, удовлетворяющего приведенным требованиям. Используя этот способ, ИЯ и ФН могут легко внедряться в самых разнородных компьютерных платформах. Он не накладывает никаких дополнительных условий и требований на разработку программного обеспечения кроме одного - ФН должно поддерживать протокол и способ взаимодействия с ИЯ. В настоящее время ведется разработка такого протокола и решение вопросов, связанных с созданием ИЯ по этому способу.

Для первой макетной реализации ИЯ был выбран более простой подход, который состоит в том, что ФН представляет собой не отдельную систему, а совокупность нескольких объектов (или, в простейшем случае, процедур), выполняющих конкретные, в том числе проблемно-зависимые функции. Такие, например, как сформировать на экране заданный вариант решения в заданной форме представления, принять и отреагировать на действие КП при непосредственном манипулировании с вариантом решения или помечивании значений и т.д. Вызов объектов ФН из ИЯ в этом случае может быть осуществлен несколькими способами. Самый простой

из них, использованный в данной реализации, - внутрипрограммный вызов процедур ФН, подключаемых к ИЯ на стадии компиляции (ИЯ и ФН в этом случае чаще всего или объектные модули, или статические библиотеки). Этот способ, очевидно, может быть использован только в локальном (несетевом) варианте СПВР. Для построения инвариантного ядра и функционального наполнения, удовлетворяющих приведенным выше требованиям, больше подходят, разумеется, другие способы реализации рассматриваемого подхода, предполагающие, что ФН подключается к ИЯ на стадии выполнения СПВР. Здесь может быть несколько вариантов реализации (например, когда ФН динамическая библиотека или удаленный сервер, содержащий процедуры ФН, вызываемые по протоколу RPC как в локальном, так и в сетевом вариантах, и др.), и они в настоящее время прорабатываются с точки зрения эффективной реализации.

В рассматриваемом случае достаточно только по заданным спецификациям ИЯ реализовать процедуры и объекты ФН. Основным недостатком этого способа является то, что соединение ИЯ и ФН становится весьма негибким, а в случае подключения к ИЯ нескольких ФН довольно сложным и запутанным.

Разработанное инвариантное ядро представляет собой модуль, реализованный в системе Turbo Pascal 7.0. В этот модуль включен объект, в котором определены все проблемно-независимые данные и процедуры, а также набор из 13 виртуальных проблемно-зависимых процедур, требующих реализации в ФН. Среди процедур ИЯ выделяются: управляющая процедура, осуществляющая управление всеми событиями, происходящими в СПВР; набор процедур получения и обработки событий от КП; набор процедур, осуществляющих формирование экранных форм, меню и диалоговых окон.

При таком построении ИЯ функциональное наполнение СПВР является подобъектом инвариантного ядра. В этом подобъекте описываются и определяются все недоопределенные в ИЯ данные и процедуры. К данным относятся два рабочих варианта решения, которые используются при работе процедур, а также несколько параметров, используемых процедурами для формирования ФПР, ФО, ФС и т.д. К процедурам, которые определяются в ФН, относятся: процедура инициализации; процедуры формирования заданного варианта решения в определенной ФПР; процедура обработки события, вызванного КП для манипулирования с компонентами решения; процедуры, реализующие достраиватели; процедуры, реализующие улучшатели; набор процедур, используемых инвариантным ядром для выполнения функций работы с вариантами решения; процедуры, реализующие формы оценивания; процедуры, реализующие формы сравнения; процедуры активизации; процедура сохранения текущего состояния СПВР.

На основе созданного ИЯ и по его спецификациям разработана СПВР

для решения конкретной проблемы оптимизации распределения работ по исполнителям, для которой даже отдельные частные постановки относятся к классу труднорешаемых задач.

### Литература

1. Хабибуллин Р.Ф. О концептуальных основах и архитектуре систем поддержки выработки решений // Инф. бюлл. Ассоц. мат. прогр. № 8. - Екатеринбург: УрО РАН, 1999. - С. 269 - 270.

2. Хабибуллин Р.Ф. Концептуальные основы и архитектура систем поддержки выработки решений // Наст. сборник.